REMARKS

Applicants respectfully request reconsideration and allowance of subject application. Claims 34-35 are canceled. Claims 1, 4, 6, 10, 11, 15, 23, and 27-31 are amended. Claims 1-33 are pending.

Applicants thank the Examiner for the detailed analysis presented in this Office Action.

Claim Rejections under 35 U.S.C. § 103

Claims 1-5, 10-12, 14-19, 22, 29, and 31-33 were rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 6,529,995 B1 to Shepherd. (hereinafter "Shepherd") in view of U.S. Patent No. 5,689,688 to Strong et al. (hereinafter "Strong"). Claims 6 and 30 were rejected under 35 U.S.C. § 103 as being unpatentable over Shepherd in view of Strong and further in view of U.S. Patent No. 5,958,019 to Hagersten et al. Claims 7 and 20 were rejected under 35 U.S.C. § 103 as being unpatentable over Shepherd in view of Strong and further in view of U.S. Patent No. 5,923,855 to Yamazaki (hereinafter "Yamazaki"). Claims 8-9 and 21 were rejected under 35 U.S.C. § 103 as being unpatentable over Shepherd in view of Strong and further in view of U.S. Patent No. 5,796,946 to Sakon (hereinafter "Sakon"). Claim 13 was rejected under 35 U.S.C. § 103 as being unpatentable over Shepherd in view of Strong and further in view of U.S. Patent No. 5,774,660 to Brendel. Finally, Claims 23-28 were rejected under 35 U.S.C. § 103 as being unpatentable over Shepherd in view of Strong and further in view of Yamazaki in further view of Sakon. Applicants respectfully traverse the rejections.

The subject application is directed to challenges facing, for example, a situation in which a variety of internet-based server applications require access to certain data. In situations where multiple servers are executing multiple instances of a particular server application, all instances of the server application require access to the same set of data. For example, a commerce-related server application requires access to a set of data containing information regarding product (or service) prices, shipping charges, and promotions or other discounts. If multiple servers are executing the same commerce-related server application, each instance of the application must access the same set of data to accurately and consistently calculate the price of a customer's order. If different instances of the commerce-related server application access different sets of data, different instances of the application may calculate different prices for the same order. To avoid this problem, it is important that all instances of a particular application access the same set of data or access different sets of synchronized data.

In the internet server application discussed above, all web servers in the web farm must access the same set of data regarding pricing, shipping, and discounts. If different web servers apply different sets of data, then the price of the customer's order may change with each new web page access. For example, a simple "web page refresh" command will cause the browser to retrieve the "refreshed" web page from the web server associated with the next IP address in the list. If the new web server applies a different set of data to determine pricing, shipping costs, and discounts, then the price displayed to the customer may change after the page is refreshed, even though the actual order has not changed. This situation is undesirable and may cause the customer to abandon the web site,

thereby resulting in lost revenue (and possibly a lost customer) for the operator of the site.

To ensure that all web servers access the same set of data, some existing systems use a two-tier approach in which multiple web servers access data stored in a database through a cache server. However, although such a system ensures that each web server accesses a common set of data, the use of a single cache server also introduces a single point of failure. Additionally, if network traffic is heavy, the time required to retrieve data a network may significantly delay the generation of a web page by a web server. If data retrieval speed is important, then the delays associated with a system in which the web servers access data stored in a database through a cache server may prevent acceptable operation.

Rejections of Claims 1-5, 10-12, 14-19, 22, 29, and 31-33

Respectfully, the combination of Shepherd and Strong, even if one of ordinary skill in the art would have thought to create such a combination, neither teaches nor suggests the methods and system recited in independent claims 1, 15, and 29, and claims depending from them. Claims 1, 15, and 29, as amended, are reproduced below for convenience:

> 1. (Currently Amended) A method of synchronizing activation of scheduled update data among a plurality of web servers, wherein each of the plurality of web servers is coupled to a common data server, the method comprising:
>
> receiving a scheduled activation time from the data server;

prior to the scheduled activation time, receiving the scheduled update ~~updated~~ data into staging caches in the plurality of web servers; and

at the scheduled activation time, activating the scheduled update data by causing the scheduled update ~~updated~~ data from the staging caches within each of the plurality of web servers to be accessible from an active cache within each of the plurality of web servers.

15. (Currently Amended) A system of synchronizing activation of scheduled update data comprising:

a plurality of web servers coupled to a common data server, wherein each of the plurality of web servers comprises:

a staging cache;

an active data cache coupled to the staging cache;

wherein the web server is configured to ~~retrieve~~ receive a scheduled activation time from the data server, and further configured to receive the scheduled update data from the data server into the staging cache prior to the scheduled activation time; and

wherein the web server is configured to cause the scheduled update data from the staging cache to be accessible from the active data cache at the scheduled activation time.

29. (Currently Amended) A method of synchronizing activation of scheduled update data among a plurality of web

servers, wherein each of the plurality of web servers is coupled to a common data server, the method comprising:

providing a scheduled activation time from the data server to each of the plurality of web servers;

communicating the scheduled update ~~updated~~ data into a staging cache in each of the plurality of web servers prior to the scheduled activation time; and

causing the scheduled update data from the staging cache in each of the plurality of the web servers to be accessible from an active cache in each of the plurality of the web servers at the scheduled activation time.

As described below, amendments to claims 1, 15, and 29 serve to clarify patentable distinctions between references cited and any other prior art.

Shepherd fails to teach or suggest the elements recited in claims 1, 15, and 29 and claims depending from them. In general, Shepherd discloses an appropriately titled "METHOD AND APPARATUS FOR ***MAINTAINING AND RESTORING*** MAPPING TABLE ENTRIES AND DATA IN A ***RAID SYSTEM***" (emphasis added). Expressly, in the first section describing the problem it seeks to solve, Shepherd's focus on maintaining and restoring old data: "It is a problem in the field of computer systems to restore data that has been accidentally or intentionally modified or deleted" (Shepherd, Column 1, Lines 29-31). In addition, from the first paragraph of its detailed description, Shepherd teaches a process for preserving superseded data to facilitate error recovery:

> The present invention provides for ***maintaining*** a set of unexpired ***modified or deleted data*** which can be used ***to recover data to a previous point in time***, with the unexpired data being

> intelligently aged (i.e., made expired) and collected by a background free space collection process. This capability is especially important for disaster recovery, and for less intrusive mistakes such as those caused by user or program error where data is wrongly modified or deleted. ***The present invention preserves the unexpired data based on one or more methods (e.g., time, available space) until it becomes expired and so becomes available for storage of new data.***

(Shepherd, Column 3, Line 60, through Column 4, Line 4; emphasis added). Furthermore, Shepherd teaches maintaining modified or deleted data in a RAID (Redundant Array of Independent Disks) system to provide data recovery between independent disks within a single computing system. In short, Shepherd's object is to preserve old data to facilitate restoration of old data to facilitate recovery from disasters or errors within a single computing system.

In its focus in preserving modified or deleted data, Shepherd, if anything, teaches away from the recitations of the claims 1, 15, 29, and claims depending from these claims. Applicants have amended claims 1, 15, and 29 to further clarify distinctions between Shepherd and any other prior art. Specifically, for example, claim 1 as amended recites "receiving scheduled update data into staging caches in the plurality of web servers" and "at the scheduled activation time, activating the scheduled update data." As previously described, Shepherd's focus is retrospective; Shepherd targets maintaining previously modified data for purposes of retroactively restoring that data to an earlier state in the event of subsequent data loss. Shepherd does not concern update data, let alone activating such data at a scheduled activation time. Respectfully, therefore, Shepherd neither teaches nor suggests what is recited by claims 1, 15, and 29 as amended.

The failure of Shepherd to teach or suggest what is recited in claims 1, 15, 29 is highlighted by considering specific passages cited in the Office Action as applied to elements of the claims. First, the Office Action cites two passages of

Shepherd as disclosing "receiving updated data into the staging caches in the plurality of web servers" as previously recited in claim 1. The first cited passage clearly describes the recovery of old, superseded data:

> The above-described problems are solved and a technical advance is achieved in the field of computer systems by the ***temporary preservation of physical space occupied by previously modified or deleted data stored in the data storage subsystem*** . . . .

(Shepherd, Column 2, Lines 9-12; emphasis added). In addition, the second passage cited concerns the transfer of a data record that may have been staged into cache memory (*see* Shepherd, Column 10, Lines 4-6 and 10-12). However, the second passage of Shepherd is extracted from a description of Figure 3 of Shepherd, which details the steps "to read data from a data ***redundancy group in the disk drive subsets***" in a RAID system (Shepherd, Column 9, Lines 64-65; reference numbers omitted; emphasis added). Respectfully, neither temporarily preserving space occupied by previously modified or deleted data nor reading data from a redundancy group in disk drive subsets teaches or suggests "receiving scheduled update data into staging caches in the plurality of web servers" as recited in claim 1. Thus, applicants submit that the cited passages of Shepherd neither teach nor suggest "receiving scheduled update data" as recited in claim 1.

Second, the Office Action cites two additional passages of Shepherd as disclosing "copying data from the staging cache of each web server to an active cache of each web server to an active cache of each server." Applicants wish to note that "copying data" previously was amended to read "causing updated from a staging ached to become accessible from an active cache." Moreover, the passages cited in the Office Action, particularly when read together, do not

disclose causing scheduled update data from a staging cache to become accessible from an active cache as recited in claim 1 as amended:

> Therefore in order to perform a write operation, the virtual track that contains the data record to be rewritten is staged from the logical layer into the cache memory.
>
> * * *
>
> When data in cache memory is modified, it cannot be written back to its previous location on a disk drive in disk drive subsets since that would invalidate the redundancy information on that logical track for the redundancy group. Therefore, once a virtual track has been updated, that track must be written to a new location in the data storage subsystem and the data in the previous location must be marked as free space.

(Shepherd, Column 11, Lines 12-17, and Column 14, Lines 24-30; reference numbers omitted). Again, the focus of Shepherd is on writing changed data records to different locations on one or more disks in a RAID system to preserve the superseded data for possible error recovery. Although Shepherd does describe data being "staged" into a "cache memory," this data is being staged to a memory to facilitate the new data being rewritten to a different disk location. Shepherd does not disclose, nor does it suggest, using an active cache in addition to the staging cache, or scheduled update data stored in the staging cache becoming accessible from the active cache. Therefore, applicants submit that these cited passages of Shepherd also fail to teach or suggest what is recited in claim 1 as amended.

In addition, applicants submit that no person of ordinary skill in the art at the time the invention was made would have been motivated to modify the teachings of Shepherd with the teachings of Strong. Shepherd discloses a RAID system in which multiple, independent storage disks are accessed within a single

system. By contrast, Strong teaches a process by network nodes are synchronized to a master node via an exchange of broadcast messages as recited in passages cited in the Office Action. (Strong, Column 2, Lines 7-15, and Column 9, Lines 51-53). In a single system with a redundant array of disks as described by Shepherd, there is no need or desirability to synchronize a system clock with another clock. Shepherd describes an error recovery process operating within a single computing system with a single system clock. Accordingly, applicants submit that it is improper to combine Strong with Shepherd.

However, for the sake of argument, even if one of ordinary skill in the art at the time the invention was made might have combined Shepherd and Strong, their combination still does not disclose what is recited in claims 1, 15, and 29. At most, Strong might be used to synchronize system clocks in a network where, as recited in claim 1, a "plurality of web servers is coupled to a common data server." Nonetheless, however, Strong fails to teach or suggest "receiving a scheduled activation time" to facilitate "at the scheduled activation time, activating the scheduled update data by causing the scheduled update data from the staging caches within each of the plurality of web servers to be accessible from an active cache within each of the plurality of web servers." In short, Strong may describe synchronizing clocks, but fails to disclose how times generated by the synchronized clocks may be used as recited in claim 1.

In sum, Shepherd describes a system for maintaining modified and deleted data to facilitate error recovery by allowing data to be restored to a former state and, therefore, teaches away from a system for causing updated information to be made available as recited in claims 1, 15, and 29. There would not have been motivation at the time the invention was made to combine Shepherd with Strong's

clock synchronization system because Shepherd concerns data redundancy in a single computing system where there are not multiple clocks to synchronize. Further, even if Shepherd and Strong were to be combined, neither teaches receiving a scheduled activation time and activating scheduled update data at the scheduled activation time as recited in claims 1, 15, and 29.

Accordingly, applications respectfully submit that claims 1, 15, and 29 are patentable over the references cited, and request that the rejection under 35 U.S.C. § 103 be withdrawn as to claims 1, 15, and 29. In addition, because claims 2-14, 16-22, and 30-33 depend from and apply additional limitations to claims 1, 15, and 29, respectively, applicants respectfully request that the rejection under 35 U.S.C. § 103 also be withdrawn as to claims 2-14, 16-22, and 30-33.

<u>Rejections of Claims 23-28</u>

Respectfully, the combination of Shepherd, Strong, Yamazaki, and Sakon, even if one of ordinary skill in the art would have thought to create such combination, neither teaches nor suggests the methods and system recited in claim 23 as amended:

> 23. (Currently Amended) One or more computer-readable media having stored thereon a computer program that when executed performs a method comprising the following steps:
>
> retrieving receiving a scheduled activation time from a data server;
>
> prior to the scheduled activation time, receiving <u>scheduled update</u> ~~updated~~ data into a staging cache in a server;

at the scheduled activation time, causing <u>the scheduled update</u> data from the staging cache in the server to be accessible from an active cache in the server; and

after the scheduled activation time, updating data caches in the data server and calculating a next scheduled activation time.

As described below, amendments to claim 23 serve to clarify patentable distinctions between references cited and any other prior art.

Not even a combination of all four references combined in the Office Action – if it would have been obvious or logical for one skilled in the art at the time of the invention to combine all four references – teaches or suggests what is claimed in claim 23. First, if anything, Shepherd teaches away from what is recited in claim 23. As previously described, Shepherd discloses an error recovery process for use in a RAID system that, while including multiple independent disks, uses a single processor. The error recovery process preserves previously modified and deleted data allowing a RAID system to be restored to previously existing state. Claim 23, by contrast, recites "causing scheduled update data from the staging cache in the server to be accessible from an active cache in the server." Thus, instead of making scheduled update data available, Shepherd makes old data available, working backwards in time and, thus, teaching away from what is recited in claim 21.

Second, as also described above, even if for the sake of argument it would have been obvious to combine Strong with Shepherd, such a combination fails to disclose what is recited by claim 23. Strong describes synchronizing system clocks of nodes in a network to a system node. Strong does not discuss receiving an activation time upon which scheduled update data from the staging cache is

made accessible from an active cache. Therefore, Strong does not make up for the shortcomings of Shepherd with regard to what is recited in claim 23. Although the Office Action's inclusion of Yamazaki and Sakon concedes that a hypothetical combination of Shepherd and Strong fails to disclose what is recited by claim 23, applicants wish to emphasize that Shepherd and/or Strong do not provide a basis over which claim 23 might be considered unpatentable in combination with the other references cited or any other prior art.

Third, applicants submit that Yamazaki fails to make up for shortcomings of the other references cited. The Office Action acknowledges that Shepherd in view of Strong fails to disclose "after the scheduled activation time, updating data caches in the data server," but claims that Yamazaki teaches "the use and advantage of updating caches in the data server after the scheduled activation time." However, while Yamazaki does address consistent updating of a cache memory, Yamazaki does not describe updating a cache memory after a scheduled activation time. In a passage relied upon in the Office Action, it is clear that Yamazaki updates cache memories upon receipt of an update instruction:

> Referring to FIG. 7, it is assumed that "0" is substituted for the variable "X" and the cache state is under the "CE" state in the processing unit #1. ***After a synchronization instruction is executed under this state,*** when the processing unit #2 refers to the variable "X", the entry containing the variable "X" is loaded into the cache memory of the processing unit #2, so that the cache state thereof becomes the "CS" state. At the same time, the cache state of the entry containing the variable "X" of the cache memory of the processing unit #1 also becomes the "CS" state.

(Yamazaki, Column 5, Lines 48-57; emphasis added). Thus, Yamazaki synchronizes cache memories when a synchronization instruction is received, not after a scheduled activation time as recited in claim 23. Thus, Yamazaki does not

make up for the shortcomings of the other references cited in teaching or suggesting what is recited in claim 23.

Finally, applicants submit that Sakon also fails to make up for the shortcomings of the other references cited. The Office Action acknowledges that Strong and Yamazaki fail to disclose "calculating the next scheduled activation time," but claims based on Sakon that "the use and advantage for scheduling the next activation time is well known to one skilled in the relevant art at the time the invention was made." Respectfully, however, Sakon discloses nothing about calculating a next activation time. In claim 1 of Sakon cited in the Office Action for disclosing scheduling the next activation time, Sakon describes calculating not a time, but a barrier value:

> 1. A barrier synchronizer of a multi-processor system having a plurality of processors and a shared region shared by each of said processors, each processor comprising:
> a lock means for exclusively locking the shared region;
> a read means for reading a value of the shared region locked by the lock means;
> an adding means for adding a predetermined value to said value to generate an added value;
> a write means for writing the added value to the shared region;
> ***a calculation means for calculating a next barrier value based on the added value, the next barrier value representing a value of the shared region at a time when all the processors will have reached a synchronized condition***;
>
> * * *
>
> 3. The barrier synchronizer of a multi-processor system according to claim 2, wherein each processor further comprises a local region, and ***wherein said calculation means calculates the next barrier value based on a value stored in the local region.***

(Sakon, Column 8, Lines 25-40 and 54-58; emphasis added). From these passages, it is clear that Sakon concerns calculation of a barrier value, not an activation time. Barrier value is defined by Sakon as "a value of the shared region

at the time that all the processors reach synchronized condition in accordance with the calculated value of the adding means." (Sakon, Column 2, Lines 2-4). In both the passages cited in the Office Action and in the definition of "barrier value," Sakon assumes that a time or point when all the processors have reached a synchronization value is known, for nothing in Sakon addresses how it is determined "when all the processors will have reached the synchronization condition," and discloses nothing as to how an activation time is calculated. In sum, Yamazaki fails to make up for the shortcomings of Shepherd, Strong, or Yamazaki.

In sum, Shepherd describes a system for maintaining modified and deleted data to facilitate error recovery by allowing data to be restored to a former state and, therefore, teaches away from a system for causing updated information to be made available as recited in claims 1, 15, and 29. There would not have been motivation at the time the invention was made to combine Shepherd with Strong's clock synchronization system because Shepherd concerns data redundancy in a single computing system where there are not multiple clocks to synchronize. Further, even if Shepherd and Strong were to be combined, neither these references nor Yamazaki or Sakon teach "updating data caches in the data server and calculating a next scheduled activation time" as recited in claim 23.

Accordingly, applications respectfully submit that claim is patentable over the references cited, and request that the rejection under 35 U.S.C. § 103 be withdrawn as to claim 23. In addition, because claims 24-28 depend from and apply additional limitations to claim 23, applicants respectfully request that the rejection under 35 U.S.C. § 103 also be withdrawn as to claims 24-28.

Applicants' decision not to discuss the differences between the cited art and each dependent claim with regard to rejections under 35 U.S.C. § 103 should not be considered as an admission that applicants concur with the Office Action's conclusion that these dependent claims are not patentable over the cited references. Similarly, applicants' decision not to discuss differences between the prior art and every claim element, or every comment in the Office Action, should not be considered as an admission that applicants concur with interpretations and assertions regarding those claims in the Office Action.

Conclusion

Claims 1-33 are in condition for allowance. Applicants respectfully requests reconsideration and prompt allowance of the subject application. If any issue remains unresolved that would prevent allowance of this case, **the Examiner is requested to urgently contact the undersigned attorney to resolve the issue**.

Respectfully Submitted,

Date: September 6, 2005

By: Frank J. Bozzo

Frank J. Bozzo
Lee & Hayes, pllc
Reg. No. 36,756
(206) 315-4001 ext. 103